External script authentication

Last updated: Tue, 07 Aug 2018 17:23:35 GMT

Overview

EZproxy has two methods that allow you to use a script on a web server for user authentication. The method described on this page configures EZproxy so that it performs all direct user interaction during authentication, then calls upon a user provided script to interpret whether or not the username and password are valid. The other option is CGI user authentication in which all aspects of handling authentication, including user interaction, must be performed by your own script.

The following are examples you might make in user.txt and what they do. In these examples, when ^u appears, EZproxy will automatically replace that with the username provided by the user on the login form; ^p will be replaced by the password from the login form; and ^i will be replaced by the IP address of the remote user. Also, when checking the "valid" string, all user-supplied strings are compared as case-insensitive.

```
::external=http://auth.yourlib.org/cgi-bin/check.cgi?user=^u&pass=^p
```

This entry specifies that EZproxy should connect to the script http://auth.yourlib.org/cgi-bin/check.cgi using a GET method and send the variables user and pass with the username and password that was provided on the login form. If this script displays either of the messages +VALID or webchkpass anywhere in its response, then EZproxy will consider the authentication to be valid.

```
::external=http://auth.yourlib.org/cgi-bin/check.cgi?user=^u&pass=^p,valid=+OK
```

This example is the same as the first, except that EZproxy will look for +OK in the response from your script to indicate the information provided is correct.

```
::external=http://auth.yourlib.org/cgi-bin/check.cgi,post=user=^u&pass=^p,valid=+OK
```

This example is the same as the second, except that EZproxy will use the POST method to submit the username and password. The POST method is preferred, since it prevents this information from being recorded into your web server log file.

Examples

- A simple ASP external script that checks a username and password against an SQL database (replace file extension with ".asp" to use script)

```
<%  
  if request("user") = "" then
  %>
  <!-- Simple form for interactive testing, not need by EZproxy -->
  <form method="post">
```

https://help.oclc.org/Library_Management/EZproxy/Authenticate_users/EZproxy_authentication_methods/External...
Username: <input type="text" name="user"><br>
Password: <input type="password" name="pass"><br>
<input type="submit">
</form>
</%
else
%

Testing
<!--Include ADO named constants -->
<!--#include file="http://www.oclc.org/common/includes/asp/adovbs.inc" -->
<%
Set cnn1 = Server.CreateObject("ADODB.Connection")
openStr = "dsn=users"
cnn1.Open openStr, "",""
sql = "SELECT distinct username from users where " & _
"username = " & Replace(request("user"), "", ") & " and " & _
"password = " & Replace(request("pass"), "", ") & ""

Set rsMbrs = Server.CreateObject("ADODB.Recordset")

rsMbrs.Open sql, cnn1, adOpenForwardOnly, adLockReadOnly, adCmdText
Do While Not rsMbrs.EOF
    response.write("+VALID")
    rsMbrs.MoveNext
Loop
' Close the recordset and discard the object.
rsMbrs.Close
set rsMbrs = Nothing
' Close the connection and discard the object.
cnn1.Close
set cnn1 = Nothing
end if
%

• A simple PHP external script that checks a username and password against a MySQL database (replace file extension with ".php" to use script)

<?php
/* mysql.php
(c) Copyright 2008 OCLC Online Computer Library Center, Inc.
http://www.oclc.org/

This is a sample PHP script that demonstrates how to perform external user validation for EZproxy. This particular example shows how to check the user's credentials against a MySQL database, although any form of relevant validation could be implemented in the check_user function.

To configure EZproxy to reference this script, make an entry similar to this in user.txt/ezproxy.usr:

::external=http://auth.yourlib.org/cgi-bin/mysql.php,post=user=^u&pass=^p
*/
/*
/* check_user is the routine where you create your own validation function. This sample routine connects to a mysql database and queries on the username to access a plain-text password from the database. If the user is found, the password is validated by simple string comparison.

check_user should return 0 for invalid user, non-zero for valid user
*/

function check_user($user, $pass)
{
    $valid = 0;

    $link = mysql_connect("localhost", "mysqluser", "mysqlpass")
        or die("Unable to connect to mysql server");
    mysql_select_db("mysqldb", $link)
        or die("Unable to select mysql database");
    /* Quote the username to make it valid to pass on to mysql */
    $quser = ereg_replace("'", "''", $user);
    $query = mysql_query("select password from users where username = '$quser'")
        or die("mysql query invalid");
    if (list ($rpass) = mysql_fetch_row($query)) {
        if (strcmp($pass, $rpass) == 0) {
            $valid = 1;
        }
    }
    mysql_close($link);
    return $valid;
}

/* Since the username will be sent to MySQL, validate it against a regular expression.
Only allow usernames that are letters, numbers, hyphens, underscores,
and in general are just 1-32 characters in length
You can change this check as needed.
*/
if (preg_match("/^[\-_a-zA-Z0-9]{1,32}$/", $HTTP_POST_VARS["user"], $regs)) {
    $user = $regs[0];
} else {
    $user = ";
}

/* Pass isn't checked since the example does not send it to MySQL, so there is no security risk. If you change that assumption, you should validate it more closely.
*/
$pass = $HTTP_POST_VARS["pass"];
if (strcmp($user, "") != 0) {
    if (check_user($user, $pass)) {
        echo("+VALID
");
    }
    /* IE gets mad if it gets nothing back at all, so send something */
    echo("Done\n");
    exit(0);
}

This form is for testing. EZproxy does not use this script, and end-users of EZproxy will never see it.

<form action="<?php echo(getenv("SCRIPT_NAME")); ?>" method="post">
    Username: <input type="text" name="user"><br>
    Password: <input type="password" name="pass"><br>
    <input type="submit">
</form>

Groups

An external script may also indicate which groups to which an EZproxy user should be given access. To do this, the script must not only output the valid message, but it must also display a line formatted like this:

```
    ezproxy_group=General+Legal
```

There may not be any white space in front of this line. In the absence of such information, EZproxy will place the user in whatever groups have been chosen in user.txt.